Development of a Scientific Visualization Tool for Vehicle Dynamics

Nancy Rowe
Graphics Application Specialist

AHPCRC
1200 Washington Ave. S.
Minneapolis, Minnesota 55415

nrowe@networkcs.com

Dan Kedziorek
Software Engineer

US Army TACOM-TARDEC
High Performance Computing Team
AMSTA-TR-VP #157
Warren, Michigan 48397-5000

kedziord@tacom.army.mil

Keywords: OpenGL, scientific visualization, High Performance Computing (HPC), vehicle dynamics, analytical simulation, software engineering

Abstract

The "DADS Interactive Visualizer from the AHPCRC" (DIVA) provides TACOM-TARDEC engineers with a customized post-processing software package designed to read and display modified output from LMS CADSI's mechanical simulation program called Dynamic Analysis and Design System (DADS), generated using TARDEC's HPC systems.

The DADS computer program is a highly detailed, general-purpose modeling and simulation method for determining the spatial, transient-dynamic response of controlled, articulated multi-body mechanical systems subject to excitation by irregular external and internal forces and disturbances. The DADS program consists of three main parts: a preprocessor, main processor, and post processor. For use with DIVA (and for purposes of this paper), we will focus on the post processing, which allows the vehicle dynamic state variables to be plotted and (when supplied with geometrical outline descriptions of the bodies comprising the system) to generate three dimensional representations for visual interpretation. DIVA further enhances this capability by allowing for the creation of animation files (i.e., MPEGs) as well as other desirable features (such as lighting control, model rotation, etc) all in a single visualization application. DIVA uses the structured output of DADS combined with a visual model of a vehicle—a geometrical outline description of the bodies comprising the system, pulled from historical databases or created as needed—to generate and save animations over time.

DIVA was developed by a software engineer and graphics application specialist using OpenGL and freely available graphics libraries and tools. Initially developed in a Silicon Graphics, Inc (SGI) IRIX (UNIX) environment, DIVA has since been ported to work in NT and LINUX as well. Although designed specifically to be used with DADS-specific output, the DIVA code—and the techniques used in its development—can be applied to just about any scientific visualization application.

This paper covers the background of TACOM-TARDEC's vehicle dynamics modeling and simulation efforts and how such a visualization tool contributes to the analysis process. Code development, focusing on the components, libraries, executables, and features of DIVA will be explored. Items for future development will be mentioned.

Background

Growing complexity of analytical simulations have opened a world of requirements and opportunities to effectively, efficiently, and convincingly use the output from supercomputer-based simulation and analysis software packages. The most natural and immediate answer is through scientific visualization. However, most of today's visualization application packages (such as Ensight or SGI's "ivview") provide general—and usually limited—functionality to satisfy the needs for a variety of uses. This paper will focus on the development and functionality of a fairly specific visualization tool for vehicle dynamic analysis.

The US Army Tank-automotive Armaments Command, Tank-Automotive Research Development Engineering Center (TACOM-TARDEC) is home to much of the United States' military ground vehicle system development and fleet maintenance. Specifically, the TACOM-TARDEC Analytical Simulation Team performs supercomputer-based simulations of the dynamics for the total vehicle system—and/or its subsystems—primarily using an application called DADS. Dynamic analyses are performed to determine feasibility of future vehicle configurations for various purposes including operational and performance specifications that contribute to the requirements document for future vehicle systems, as well as during the source selection process to determine best performance evaluations. By coupling the dynamic analysis capabilities and with Finite Element Analysis (FEA) capabilities, potentially dangerous and costly field testing can be simulated in the computer reducing costs and minimizing risks to life and equipment (i.e., a vehicle roll-over or clearance of a gun tube when traversing a ditch). Although these simulations are no substitute for field-testing they have shown significant reduction in risks and costs.

TACOM-TARDEC engineers use dynamic analysis and scientific visualization applications to closely scrutinize the dynamic performance of vehicles while they are still in the concept stage of development, to directly and significantly impact the design process, and to contribute to modifications and sustainment throughout the vehicle's life cycle. Additionally, the information gained from computer-based simulations and computer-generated animations is beginning to influence the dynamic operational and performance specifications which go into the requirements document for future vehicle systems.

TACOM-TARDEC is also home to one of several High Performance Computing (HPC) Distributed Centers within the Department of Defense (DoD) HPC Modernization Program. As a Distributed Center, the TARDEC HPC Team maintains a unique local requirement of integrating the latest high-end supercomputing technology for use in real-time man/hardware-in-the-loop simulations of military ground vehicle systems. This mission also includes the use of centralized HPC resources to promote scientific visualization efforts, specifically in virtual prototyping and immersive environments. It is through the inclusion of TARDEC in the HPC Modernization Program that application development, such as DIVA, is possible.

DIVA Development

Working with a software engineer on the TARDEC HPC Team, a graphics application programmer from the Army High Performance Computing Research Center (AHPCRC) developed DIVA to meet specifications from end-user TACOM-TARDEC engineers. A "modified waterfall" approach to software development was used where several iterations between the developer and end-user were required.

Needed was a visualization application capable of reading data generated by the DADS mechanical analysis software for both wheeled and tracked vehicles. The application had to be capable of displaying the movements of a vehicle over time using the DADS parameter file which determines the position of the vehicle model at each timestep. Major output requirements included the ability to save images and animations, to interact with the vehicle using a mouse (or other navigable device), and to change colors of vehicle components—all in real time.

Initially, the targeted platform for DIVA was an SGI supercomputer-class system or SGI workstation. All development of DIVA was done using an SGI O2 system. As development progressed, the idea of cross-platform usability through OpenGL was considered. DIVA was then compiled in a LINUX environment to ensure that the code ran correctly on both LINUX and SGI IRIX operating systems. After the initial development of DIVA was completed on the SGI system, DIVA was ported to an NT environment as well.

DIVA components

DIVA is composed of two main parts, the Graphical User Interface (GUI) and the application code. The GUI, providing the general functions for DIVA, was built using the Viewer Framework code developed at AHPCRC while the application portion of DIVA handles the reading of data (in a particular format generated by DADS) and the rendering of a real-time image (as much as 30 frames per second).

The Viewer Framework provides the capability to rapidly build an application that can display 2D and 3D graphical data. The Viewer provides the ability to view a model from different directions, focus on particular parts of the model, display multiple simultaneous views of the model, display the model using various graphical display techniques, animate a time-dependent model, save still images (frames) or movies of the model, and other features typically found in graphical viewing applications. Although not a stand-alone all-encompassing simulation application, DIVA provides a complete suite of visualization tools and techniques to be used in conjunction with a numerical analysis package.

The application portion of DIVA reads and displays the data, consisting of several files.

- The DADS Parameter file contains the positional and rotational data for each body in the model at each time step. This file is created as a direct output file by a DADS simulation.
- The Color file specifies the color for each body. This file can be modified to highlight certain parts of a vehicle system. Or, when used as a "color palette", can be reused in other simulations/visualizations.

- The Geometry files contain the geometry information for each body in a model. DIVA Geometry files are created by converting DADS Geometry files into “Movie.BYU” format or by converting a geometry file created by a modeler such as Alias/Wavefront's Maya into Movie.BYU format. The Movie.BYU format originated at Brigham Young University. Movie.BYU format describes a model as an unstructured collection of polygons. Movie.BYU format is commonly used to display surfaces of objects.
- The List file simply contains a list of the DIVA Geometry files for the data set.
- The Set file is essentially a startup file for DIVA. It contains the filenames of the Parameter file, the Color file, the List file as well as a list of which bodies are to be displayed. Often only a subset of the bodies used in analysis are displayed. (For example, although a DADS simulation may include under-body suspension components to achieve the vehicle’s overall dynamics, the engineer may only be interested in visualizing how the cab, crew compartment, or cargo will react to those dynamics. Consequently, the suspension components need not be visualized. Of course, if the movement of specific suspension components is under analysis, then the visual of the cab, crew compartment, or cargo will not even be modeled).

Portability

A major goal in designing DIVA and the Viewer Framework was portability. To achieve portability, it was necessary that the libraries used by DIVA be free, non-proprietary, stable, commonly used, well supported, and have a C binding. Libraries supporting the X11 protocol were chosen so that the DIVA application would run on an SGI supercomputer and be accessible by NT or other high-end workstations through a graphical X-window session. The libraries used include OpenGL, Tcl/Tk, and ImageMagick.

OpenGL

Graphic functions and routines in DIVA are written using the OpenGL Application Programming Interface (API). OpenGL is currently perhaps the most widely used and most common library for developing portable 3D graphics applications. The OpenGL Architecture Review Board, an independent consortium made up of representatives from the leading graphics vendors, determines the OpenGL specification thereby ensuring support across multiple platforms. OpenGL is a stable environment, having been available on many platforms for more than eight years. Of course, OpenGL continues to evolve to take advantage of new developments in hardware and graphic architectures. OpenGL is well documented in books and through the http://www.opengl.org website.

Tcl/TK

The DIVA GUI is written using Tcl/Tk. Tcl (Tool command language) and Tk (Toolkit) were written by John Ousterhout. Together they provide a programming system for building X Window interface applications. Tcl is a simple scripting language. Tk provides facilities to build Motif-like interfaces. Tcl and Tk are “open source”, widely used and highly portable. On-going development and support is provided by engineers at Scriptics and by the large Tcl/Tk user community. Rapid interface development is possible with Tcl because it is an interpreted language. Because no compilation of the code is required development can be done interactively.

ImageMagick™

Through the Viewer Framework, DIVA makes function calls to ImageMagick™ when generating images and animations. ImageMagick™ is a freely available collection of tools and libraries for image manipulation. Capable of writing over 68 major image formats, this functionality gives DIVA its flexibility and versatility in generating output formats. Since ImageMagick™ is actually a separate component from DIVA, it must be resident on the system where DIVA is running. Although DIVA will run without ImageMagick™ installed, without ImageMagick™ the ability to write images and animations will be disabled.

Tcl/Tk, OpenGL, and ImageMagick™ are currently supported on Unix, Linux, and NT. All of these products are available free of charge and can be downloaded from the web.

Standalone executable

Mktclapp is used to create DIVA as a standalone executable. Developed by D. Richard Hipp at Hipp, Wyrick & Company, Inc. (Hwaci), Mktclapp is a utility that simplifies the task of building a program that uses both C/C++ code and Tcl/Tk. Mktclapp enables C code to call Tcl code and conversely Tcl code to call C code. Primary reason for using Mktclapp was to build a DIVA executable that did not require the Tcl/Tk libraries to be installed.

The advantage of using Mktclapp to create a standalone executable is that DIVA can be distributed to users without having to ensure that the correct and necessary libraries are installed on the system where DIVA will reside. Mktclapp converts Tcl/Tk scripts into static C strings to be linked into the executable at compile time. Also, C functions can be registered as Tcl/Tk commands so that the Tcl/Tk scripts can directly execute those C functions. Mktclapp can be configured to link in all the static libraries for Tcl/Tk.

xmktclapp.tcl, the GUI wrapper for Mktclapp, makes Mktclapp even easier to use. Using Mktclapp's GUI, the C and Tcl files, as well as the location of the Tcl and Tk libraries that DIVA uses, are specified. After specifying that the executable should be "standalone", the "Build" command is issued to create corresponding *.c and *.h files. Those *.c and *.h files are then compiled to build a standalone executable.

Mktclapp and its companion GUI wrapper are available free of cost at http://www.hwaci.com. Mktclapp and the GUI wrapper are licensed under the GNU Public License. More information on the GNU Public License may be found at http://www.gnu.org/copyleft/gpl.html.

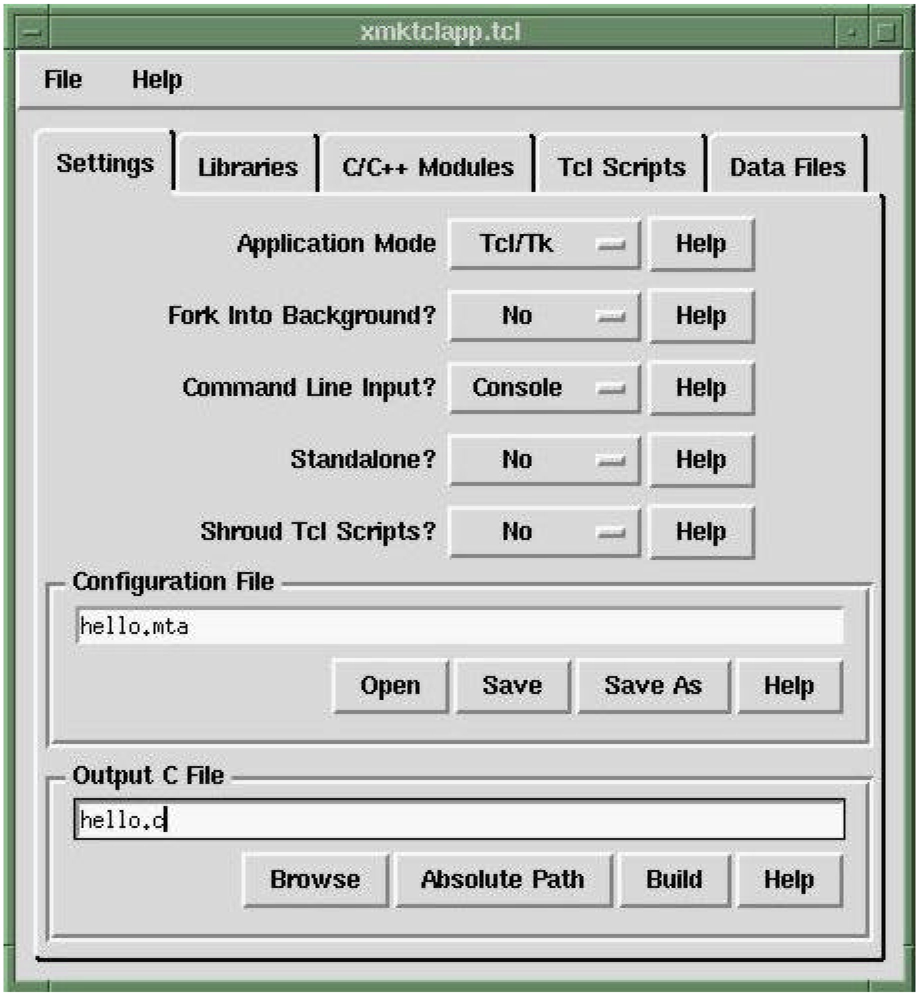

Figure 1. Snapshot of xmktclapp.tcl

Features of DIVA

- Optional mouse or command line interface
- Morphing between cameras
- Multiple layout options
- Perspective or orthographic display
- Choice of lighting
- Fast motion drawing
- Ability to save images in more than 60 formats
- Ability to create MPEG animations
- Context sensitive help
- Single frame animation
- Depth cueing
- Backface culling

- 3 animation modes
- Ability to change part colors
- Ability to undo/redo changes
- Choice of display style – points, wireframe, flat shaded, smooth shaded
- The ability to see up to four cameras at one time
- Ability to unmap/remap the GUI

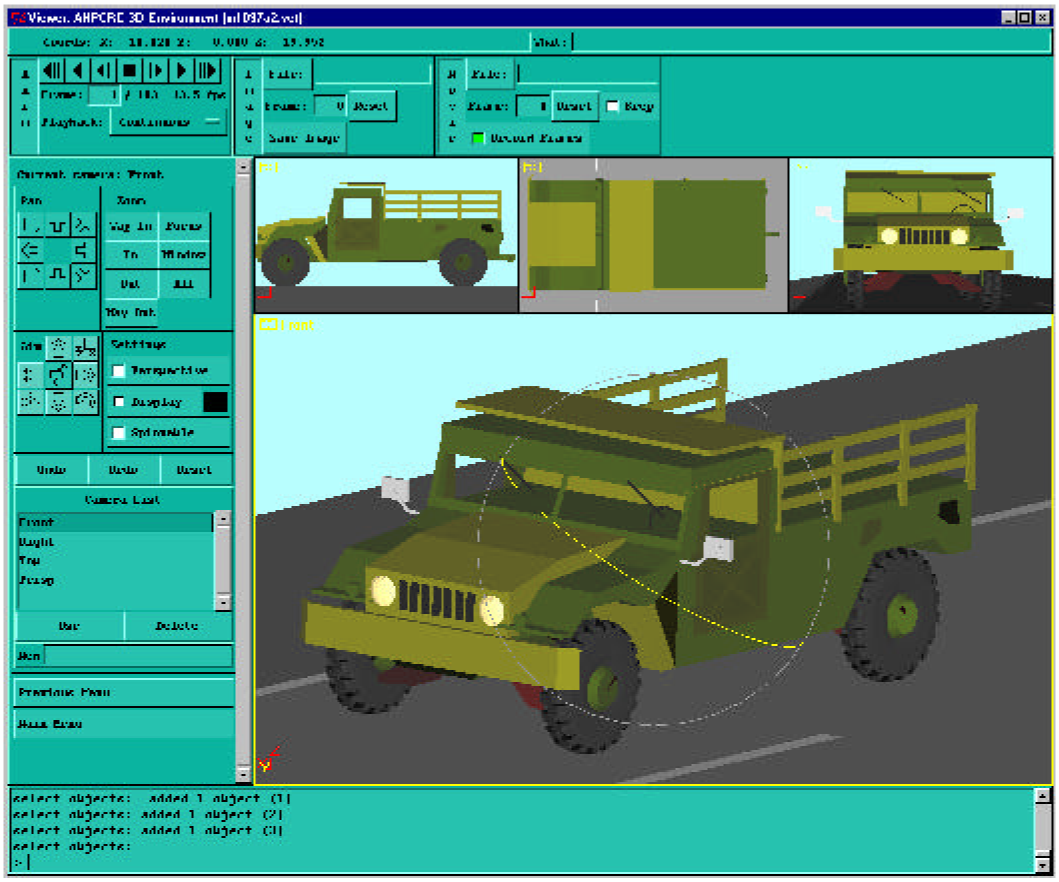

Figure 2. Screenshot of DIVA showing four simultaneous camera views and GUI interface.

Future Plans

Although the current DIVA application was developed to replace the basic functionality of an older (currently no longer supported) GL-based visualization application, incorporation of additional features into DIVA is planned. These include texturing, transparency and the ability to look at multiple data sets at one time.

The file formats for DIVA are also based on historical applications. Future plans include streamlining these file formats.

The NT port of DIVA is another area that requires additional work. The NT version currently lacks some of the features of the more-capable IRIX version of DIVA. Because of the nonavailability of a tool similar to “Mktclapp” for NT, a standalone NT executable cannot be generated. As a result, all the components (including ImageMagick, Tcl/Tk, etc.) must be installed and configured on the NT machine. The NT version also needs additional integration of the ImageMagick library.

Acknowledgement

This work was sponsored in part by the Army High Performance Computing Research Center under the auspices of the Department of the Army and the Department of Defense HPC Modernization Program. The content does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

References

http:/www.tacom.army.mil/hpc